

Trabalho Prático 1 - Simulador NEANDER

Escrever um programa para o simulador Neander que calcule o valor da função “soma dos divisores positivos” $\sigma_x(n)$ para um valor de entrada n e uma potência x . A função divisor é definida como:

$$\sigma_x(n) = \sum_{d|n} d^x$$

Nesta definição $d|n$ significa “ d é divisor de n ”, ou, em outras palavras, o resto da divisão de n por d é zero. A função $\sigma_x(n)$ calcula, portanto, o somatório de todos os divisores de um número n elevados à potência x . Alguns casos interessantes são observados para $x=0$ e $x=1$. Para $x=0$, a função retornará a quantidade de divisores de n , uma vez que $d^0=1$ para qualquer d . Para $x=1$, a função retornará o somatório dos divisores de n , uma vez que $d^1=d$ para qualquer d . O programa receberá os valores de n e x como entradas e produzirá o resultado de $\sigma_x(n)$.

Esta é uma função importante em um ramo da matemática conhecido como teoria dos números, que estuda números inteiros e suas propriedades. Ela pode ser usada para definir categorias de números como números primos, perfeitos, multiperfeitos e superperfeitos.

Os dados de entrada e os resultados devem ocupar, obrigatoriamente, as seguintes posições de memória:

Palavra 128 – valor de ' n '; dado de entrada, positivo, de 1 a 127

Palavra 129 – valor de ' x '; dado de entrada, positivo, de 0 a 127

Palavra 130 – valor de $\sigma_x(n)$; dado de saída, em complemento de 2

Não há necessidade de testar os valores de entrada, pois é garantido que eles serão valores positivos em complemento de 2, conforme especificado acima.

O programa será corrigido utilizando um corretor automático, que verificará o resultado produzido pelo programa com 20 casos de teste, com diversos valores de ' n ' e ' x '. A correção é feita da seguinte forma:

1. O programa a ser testado é carregado na memória do simulador Neander (apenas uma vez).
2. Para cada caso de teste, sem recarregar o programa na memória do simulador, o corretor automático faz o seguinte:
 - Coloca 0 no PC
 - Coloca os valores de ' n ' e ' x ' nas palavras 128 e 129
 - Executa o programa que está sendo testado
 - Verifica se o valor do resultado (palavra 130) está correto ou não
 - Verifica se os valores de ' n ' e ' x ' nas palavras 128 e 129 continuam iguais aos armazenados inicialmente

Para evitar problemas durante a correção, na codificação dos programas devem ser observadas rigorosamente as seguintes especificações:

- o código do programa deve iniciar no endereço 0 da memória.
- a primeira instrução executável deve estar no endereço 0.
- os endereços dos dados de entrada e do resultado devem ser exatamente os especificados acima.
- usar para variáveis adicionais ou para código extra os endereços de memória de 131 em diante.
- os valores de ' n ' e ' x ' (endereços 128 e 130) não devem ser modificados durante a execução do programa. Ou seja, se necessário, deve-se copiar os valores de ' n ' e ' x ' para variáveis de trabalho (no endereço 131 ou superior) e codificar o algoritmo usando estas variáveis de trabalho.

Dica: o Neander não possui instrução para divisão. Esta pode ser feita por subtrações sucessivas. Similarmente, a exponenciação pode ser feita por multiplicações sucessivas que, por sua vez, podem ser feitas com somas sucessivas.

O trabalho deverá ser entregue através do sistema Moodle, na área de “Entrega do Trabalho Neander”, na forma de um arquivo compactado (formato Zip ou Rar) contendo:

- um arquivo de memória do Neander (**.mem**), com o código de máquina do programa.
- um arquivo texto, contendo o código do programa em formato simbólico, comentado (dica: usar a função “Arquivo ... Salvar texto ...” do simulador para gerar o texto inicial). Lembre-se de incluir seu nome completo e seu número de cartão nas primeiras linhas deste arquivo.
- Para dar nomes aos arquivos, utilize o seu nome completo, sem espaços e sem acentos, seguido do seu número de cartão, sem zeros à esquerda. Por exemplo: João da Silva, cartão 00123456 utilizará JoaodaSilva123456.mem, JoaodaSilva123456.txt e JoaodaSilva123456.zip (ou .rar).

A entrega de arquivos cujos nomes não obedecem a esta regra implicará em um desconto de 5% na nota do trabalho.

IMPORTANTE: Este é um trabalho **individual**. Trabalhos copiados serão duramente penalizados.

Data de Entrega: 14/10/2018 às 23h55, via Moodle. Não haverá prorrogação deste prazo.

Exemplos de casos de teste:

n (128)	x (129)	resultado (130)
5	1	6
2	6	65
12	0	6
11	1	12
11	2	122
5	3	126
49	1	57
103	0	2
1	10	1